

Modelowanie roślin przy użyciu języków formalnych

1. Wstęp

Praca dotyczy modelowania trójwymiarowego w grafice komputerowej. Jest ona propozycją nowego systemu do generowania struktur trójwymiarowych. Podstawę teoretyczną przedstawianej aplikacji stanowią L-systemy.

L-system jest językiem formalnym, stworzonym w 1968 roku przez węgierskiego biologa Aristida Lindenmayera. Lindenmayer stworzył L-systemy, podejmując próbę formalnego opisu zjawisk i zależności międzykomórkowych zachodzących w tkankach sinicy *Anabaena Catenula*, zaliczanej do królestwa bakterii oraz innych alg. Początkowo rolą L-systemów było uchwycenie topologii tkanek, a geometryczne aspekty były zaniedbywane. Pojedyncze znaki wygenerowanych słów odpowiadały komórkom. Dopiero później została zaproponowana geometryczna interpretacja i L-systemy zaczęły być używane do modelowania wyższych roślin – w tym przypadku jeden symbol oznacza już fragment łodygi, liść itp.

W przedstawianym systemie są wykorzystywane klasyczne L-systemy, parametryczne L-systemy oraz kilka zdefiniowanych przeze mnie mechanizmów.

2. L-systemy

Języki formalne bazują na idei systemów przepisujących. Podstawową zasadą działania tych ostatnich jest zastępowanie fragmentu analizowanego obiektu, który został zidentyfikowany jako występujący w zbiorze reguł poprzednik produkcji, odpowiadającym mu innym obiektem z tego zbioru: następnikiem. Tak opisany system przepisujący jest podstawą do definiowania gramatyk Chomskiego, dobrze zbadanych i szeroko stosowanych we współczesnej informatyce. L-systemy odróżnia od nich fakt, że poprzednik produkcji zawsze jest pojedynczą literą, oraz sposób określania bezpośredniego przekształcania słowa: wszystkie możliwe produkcje są stosowane równolegle, a nie jak w przypadku gramatyk struktur frazowych – sekwencyjnie. Ta różnica wynika z chęci odzwierciedlenia charakteru rozwijających się roślin: zarówno na poziomie komórkowym, jak i na poziomie złożonych części roślin wzrost następuje równolegle. Ma ona znaczny wpływ na generowane języki.

0L-systemem (L-systemem bezkontekstowym) nazywamy trójkę:

$$G = (A, \omega, P)$$

gdzie:

- A – skończony zbiór – alfabet sytemu
- $\omega \in A^+$ – symbol startowy
- $P \subset A \times A^*$ – skończony zbiór produkcji

Dla przykładu rozważmy następujący L-system:

$A = \{A, B\}$ (alfabet), $\omega = A$ (symbol startowy), $P = \{A \rightarrow B, B \rightarrow AB\}$ (dwie produkcje)

Zaczynamy od symbolu A , po zastosowaniu produkcji 1 mamy B , po zastosowaniu produkcji 2 mamy AB i dalej BAB , następnie $ABBAB$ itd.

3. Interpretacja znaków – grafika żółwia

Aby uzyskać obraz, ciągi znaków wygenerowane przez L-systemy są interpretowane za pomocą tzw. grafiki żółwia (jak w języku programowania LOGO). Rysowanie jest realizowane przez żółwia, który pamiętając cały czas swoją pozycję i kierunek, reagując na wydawane mu instrukcje, przesuwa się, jednocześnie rysując linie. Żółw jest używany do interpretacji L-systemów. Stan żółwia jest trójką:

$$(x, y, \alpha)$$

gdzie:

- (x, y) to współrzędne na płaszczyźnie,
 - α reprezentuje kierunek aktualnie uznawany jako przód.
- Dodatkowo żółw jest charakteryzowany przez dwa parametry:
- d : długość linii rysowanej w pojedynczym kroku,
 - δ : kąt, o jaki obraca się podczas obrotów.

Żółw, w podstawowej wersji, reaguje na następujące komendy:

- F : przesunąć się do przodu, rysując linię,
- f : przesunąć się do przodu, bez rysowania linii,
- $+$: skrócić w lewo o kąt δ ,
- $-$: skrócić w prawo o kąt δ .

Jako przykład działania grafiki żółwiowej na rycinie 1 przedstawiony jest smok Heighwaya (*dragon curve*):

L-system:

$$\omega = FX$$

$$P: X \rightarrow X + YF +$$

$$Y \rightarrow -FX - Y$$

Parametr δ wynosi 90° .

4. Klamrowane L-systemy

Chcąc modelować rośliny, napotykamy potrzebę wzbogacenia języka symboli interpretowanych przez żółwia o operacje na stosie. Wynika to z dominacji w świecie roślin struktur gałęziastych, które łatwiej się rysuje, zapamiętując aktualną pozycję i wracając do niej po skończeniu rysowania gałęzi.

Do poleceń żółwia dodajemy dwa symbole:

-] : włoż aktualny stan żółwia na stos,
- [: zdejmij stan z wierzchołka stosu i uczyn go aktualnym stanem żółwia.

Korzystając z powyższego, można rysować kształty przypominające naturalne rośliny. Dla przykładu przedstawię L-system pochodzący z pracy Prusinkiewicza i Lindenmayera [1990] (rycina 2):

L-system:

$\omega = F$

P: $F \rightarrow FF[-F+F+F][+F-F-F]$

Parametr δ wynosi 22° .

Iteracja : 4

5. Rysowanie w trzech wymiarach

Chcąc modelować przestrzenne rośliny, należy rozszerzyć interpretację L-systemów do trzech wymiarów. W tym celu trzeba uogólnić stan żółwia oraz jego polecenia. Poszerzamy zbiór instrukcji o obroty wokół dwóch dodatkowych osi w prostokątnym układzie współrzędnych: &, ^, \, /. Wzbogacony o rysowanie w trzech wymiarach automat może już służyć do generowania realistycznie wyglądających obrazów (rycina 3). Wystarczy go wzbogacić o symbol rysujący zdefiniowany wcześniej obiekt liścia. Przedstawiona obok roślina jest wygenerowana przy użyciu dotychczas zdefiniowanych metod. Jest to L-system działający według prostej logiki: łodyga, gałąź na lewo, łodyga, gałąź na prawo, łodyga:

$F \rightarrow F[+F]F[-F][F]$,

poszerzony o rysowanie liści i obroty wokół pionowej osi.

L-system (rycina 3):

$\omega = F$

P: F

$\rightarrow F//[///// -p] [-Fp]F//[///// -p] [-Fp][F]$

Parametr δ wynosi $35,7^\circ$.

Iteracja : 4

6. Parametryczne L-systemy

Parametryczne L-systemy okazują się niezwykle przydatnym narzędziem, ułatwiającym rozwiązanie problemów, które napotyka się, korzystając z klasycznych L-systemów. Operują one na ciągach modułów składających się z liter, należących do alfabetu V i parametrów będących liczbami rzeczywistymi. Moduł składający się z litery $A \in V$ i parametrów $a_1, a_2, \dots, a_n \in \mathbb{R}$ jest oznaczany $A(a_1, a_2, \dots, a_n)$. Będące konkretnymi liczbami parametry aktualne odpowiadają symbolicznym parametrom formalnym użytym podczas definicji L-systemu.

Parametryczny 0L-system definiuje się jako uporządkowaną czwórkę:

$$G = (V, \Sigma, \omega, P)$$

gdzie:

- V – alfabet systemu
- Σ – zbiór parametrów formalnych
- $\omega \in (V \times \mathbb{R}^*)^+$ – niepuste parametryczne słowo – symbol startowy
- $P \subset (V \times \Sigma^*) \times C(\Sigma) \times (V \times E(\Sigma))^*$ – skończony zbiór produkcji

Symbole „:” i „ \rightarrow ” są używane do rozdzielania trzech składników produkcji: poprzednika, warunku i następnika.

Przykładowa produkcja o poprzedniku $A(t)$, warunku $t > 5$ i następniku $B(t + 1)CD(t * 0.5, t - 2)$:

$$A(t) : t > 5 \rightarrow B(t + 1)CD(t * 0.5, t - 2)$$

Zastosowanie produkcji ma miejsce, gdy są spełnione trzy następujące warunki:

- litera w module jest równa literze w poprzedniku produkcji,
- liczba parametrów aktualnych w module jest równa liczbie parametrów formalnych w następniku produkcji,
- po podstawieniu parametrów aktualnych na miejsce formalnych warunków występujący w produkcji jest spełniony.

Spełniające powyższe warunki produkcja jest stosowana, tworząc nowy ciąg modułów zdefiniowany przez jej następnik, w którym odpowiednie parametry aktualne są dopasowywane do formalnych, a za ich pomocą obliczane ewentualne wyrażenia. Na przykład produkcja:

$$A(t) : t > 5 \rightarrow B(t + 1)CD(t * 0.5, t - 2)$$

w przypadku napotkania modułu $A(6)$ zostałaby zastosowana, czego wynikiem byłby ciąg:

$$B(7)CD(3, 5)$$

7. Ewaluacja wyrażeń

Parametryczne L-systemy narzucają potrzebę włączenia do programu fragmentu kompilatora, który będzie ewaluował:

- wyrażenia logiczne zawarte w warunku, do wartości prawda/fałsz,
- wyrażenia arytmetyczne zawarte w trzeciej części do wartości liczbowych.

Do tego celu została stworzona gramatyka, która definiuje języki dozwolonych wyrażeń. Wymaga to także stworzenia prostego kompilatora – użyłem do tego celu narzędzia ANTLR [ANTLR, 2009].

8. Parametryczne komendy żółwia

Parametryzacja L-systemów daje, oprócz większej swobody w definiowaniu struktury generowanych obiektów, możliwość uzależnienia komend grafiki żółwiowej od parametrów. W moim systemie zostały dodane parametryczne komendy. Oto kilka z nich:

- $+(x) - a \in \mathbb{R}$, obrót o kąt x ,
- $+(x,y) - x,y \in \mathbb{R}, y > x: +(a)$, gdzie a – losowa liczba z przedziału $\langle x,y \rangle$,
- $p(x) - x \in \mathbb{R}$: rysowanie liścia zdefiniowanego jako p , w skali x .

9. Grubość łodygi

Podczas modelowania roślin, zwłaszcza drzew, zachodzi potrzeba zróżnicowania grubości łodygi, aby zmniejszała się ona, w naturalny sposób, od podstawy do czubka. Parametryczne L-systemy pozwalają manipulować grubością, zmieniając parametr s w parametrycznej komendzie $F(d,s)$. Skorzystam tu z obserwacji poczynionej już przez Leonarda da Vinci: „Wszystkie gałęzie drzewa, na każdym etapie jego wzrostu, kiedy złączone razem, są równe w grubości pniu poniżej ich” [Richter 1970]. Więc jeśli chcemy, aby w jakiejś produkcji gałąź o promieniu r rozdzielała się na dwie nowe o takim samym promieniu s , otrzymujemy równanie:

$$\pi r^2 = \pi 2s^2$$

Stąd:

$$s/r = 1/\sqrt{2} \approx 0,707$$

Mechanizm zmiany grubości łodyg jest użyty w dalszych przykładach.

10. Zrównoważanie drzew w trzech wymiarach

Oczywiście w naturze zdarzają się asymetrie spowodowane np. nierównomiernym dostępem do światła, jednak w tym rozdziale przedstawię sposób uzyskania pewnej symetrii. Podstawową ideą jest dodanie do wierzchołka informacji o tym, jak bardzo roślina zboczyła w jedną lub w drugą stronę, i uwzględnianie tych informacji przy tworzeniu następnych rozgałęzień. Do informacji przenoszonych przez wierzchołki dodajemy cztery parametry odpowiadające za rysowanie w czterech ćwiartkach płaszczyzny. Z parametrami są skojarzone wartości odpowiadające za prawdopodobieństwo wylosowania kolejnych gałęzi w poszczególnych ćwiartkach. Gdy tworzona jest gałąź w jednej z ćwiartek, z przedziału odpowiadającego tej ćwiartce losowany

jest kąt, pod którym gałąź będzie skierowana. Następnie parametr dla tej ćwiartki zmniejsza się, a dla przeciwległej zwiększa. Pozostałe dwa zostają bez zmian:

$A(x,y,l,p,g,d)$: ($p > \%$) i inne warunki

→ gałąź w prawej górnej ćwiartce $A(x,y*0.707,l+30.0,p-30.0,g,d)$

gdzie % oznacza losową liczbę z przedziału $<0,99>$.

Stopień równomierności w rozgałęzianiu można modyfikować, zmieniając początkowe wartości parametrów oraz stałe do nich dodawane. Można również ustawić różne wartości różnym ćwiartkom i przesuwając środek ciężkości rośliny w różne strony.

11. Użycie zmiennej czasu

W bardziej skomplikowanych L-systemach pojawia się potrzeba uzależnienia aplikowanych produkcji od czasu. W naturze rozwój rośliny również następuje etapami – w pewnych porach roku rosną liście, w pewnych kwiaty, zmieniające się później w owoce. Dodałem do systemu ogólnie dostępną w produkcjach zmienną czasu t , której wartość jest równa aktualnej iteracji.

Korzystając ze zmiennej czasu oraz nowego symbolu o (owoc), stworzyłem L-system generujący jabłoń. Występujące w nim parametry odpowiadają za generowanie struktury rozgałęzień oraz modyfikacje grubości łodygi. Zmienna czasu dzieli generowanie rośliny na poszczególne fazy: najpierw generowana jest struktura, potem dodawane są cieńsze gałęzie, na koniec owoce (rycina 4).

Podsumowanie

Na podstawie przedstawionej teorii został stworzony system Turtle Painter, który daje użytkownikowi złożone narzędzie do generowania trójwymiarowych struktur. Wszystkie przedstawione w tej pracy obrazy są wygenerowane przy użyciu tego systemu. Jest on napisany w Javie, interfejs okienkowy jest stworzony przy użyciu biblioteki Swing. Do rysowania w trzech wymiarach została użyta Java 3D w wersji 1.5.0 [Java 3D 2009].

Ewaluacja wyrażeń logicznych i arytmetycznych jest realizowana za pomocą ANTLR (ANother Tool for Language Recognition) [ANTLR 2009]. Jest to narzędzie umożliwiające generowanie kompilatorów, analizatorów leksykalnych i translatorów na podstawie plików zawierających gramatyki.

Tworząc trójwymiarowe obiekty użyte w systemie, posłużyłem się Art of Illusion [Art of Illusion 2009]. Jest to program do tworzenia grafiki trójwymiarowej udostępniany na zasadzie wolnego oprogramowania.

Przy wnikliwej obserwacji roślin w naturze możliwe jest zdefiniowanie odpowiednich gramatyk oraz symulowanie za ich pomocą konkretnych gatunków roślin. Jest wiele nowych funkcjonalności, które można by dodać do systemu, aby zwiększyć jego możliwości (np. dodanie stałego wektora, który przesuwałby każdy obiekt przy rysowaniu).

waniu, symulowałyby grawitację lub wiatr), ale nawet bez nich, jak pokazują wygenerowane przykłady, możliwe jest generowanie estetycznych i realistycznych roślin.

BIBLIOGRAFIA

- ANTLR (2009). [Dostęp: 2009-03-12] Dostępne w Internecie: <<http://wwwantlr.org/>>
- Art of Illusion (2009) [Dostęp: 2009-03-12] Dostępne w Internecie: <<http://aoi.sourceforge.net/>>
- Java 3D (2009) [Dostęp: 2009-03-12] Dostępne w Internecie: <<http://java.sun.com/javase/technologies/desktop/java3d/>>
- Prusinkiewicz P., Lindenmayer A. (1990). *The Algorithmic Beauty of Plants*. New York: Springer Verlag.
- Richter J.P. (1970). *The Notebooks of Leonardo da Vinci*. New York: Dover Publications.